

文章编号: 2095-2163(2023)01-0025-06

中图分类号: TU248.3

文献标志码: A

基于分离障碍物的地下车库车位排布算法

冯嘉宇, 王潇霆, 沈 炜

(浙江理工大学 信息学院, 杭州 310018)

摘要: 地下车库中的障碍物是车位排布中除不规则外轮廓边界之外最复杂的约束条件, 为此本文提出了一种根据障碍物分离处理的启发式算法。根据障碍物分布, 将整个车库区域分为带障碍物和带不带障碍物两部分, 对其分别使用不同的方法计算。其中, 包含障碍物部分通过像素分割构建用于描述空旷程度的矩阵, 并计算最优排布方案; 不包含障碍物的部分通过遗传算法处理, 同时考虑两部分结合部位的车位损失, 并合理规划承重柱, 保证连通性。通过对不同形状、内部结构的地下车库的排布, 该算法可以在有限运算时间内得到排布结果, 为人工设计师提供参考辅助。

关键词: 地下车库; 车位排布; 像素分割; 矩阵; 栅格化

Parking space arrangement algorithm of underground garage based on separated obstacles

FENG Jiayu, WANG Xiaoting, SHEN Wei

(School of Information, Zhejiang Sci-Tech University, Hangzhou 310018, China)

[Abstract] The obstacle in the underground garage is the most complex constraint condition in parking space arrangement except for irregular outer contour boundary, so this paper proposes a heuristic algorithm based on obstacle separation. In this paper, the whole garage area is divided into two parts with and without obstacles according to the distribution of obstacles, which are calculated by different methods. The part containing obstacles is divided by pixels to construct a matrix to describe the degree of emptiness and calculate the optimal layout scheme. The part without obstacles is processed by genetic algorithm. Meanwhile, the parking loss of the part combining the two parts is considered, and load-bearing columns are planned reasonably to ensure connectivity. Through the layout of underground garages with different shapes and internal structures, the algorithm can obtain the layout results in limited operation time, which can provide reference for human designers.

[Key words] underground garage; parking space arrangement; pixel segmentation; matrix; Rasterize

0 引言

城市车位短缺是制约现代城市发展的新问题^[1]。地下车库为缓解车位短缺提供了一个成本更低的方案^[2], 广大开发商也开始研究如何在地下车库中排布才能得到更多的车位^[3], 以获取更高的经济效益。目前, 地下车库的车位排布仍以人工设计为主, 但是由于外边界不规则和内部的障碍物等问题^[4], 人工设计的效率较低。因此, 一些专家学者开始将自动化设计运用于解决地下车库的车位排布问题。

车位排布算法和工业生产中的二维排样算法很相似。早在1981年, Fowler等^[4]就已证明了不带约

束条件的排样问题是 NP 完全问题。因此, 在排样问题基础上添加了更多约束条件的车位排布问题也是一个 NP 完全问题。对于车位排布问题, 现存的大多数算法都是从传统排样问题中优化得到的。Abdelfatah等^[5]提出了一种适用于矩形区域的设计线性规划模型, 并考虑了车道夹角; Kong等^[6]通过调整车位的长和宽, 运用混合整数非线性模型计算车位长和宽的变化对于最终可排列车位数量的影响; Huang等^[7]设计了一套基于模拟退火算法的矩形区域内车位排布问题的通用算法; Syahrini等^[8]则通过线性规划模型设计了一套适用于三角形区域的车位排列算法; Ihda等^[9]提出了一种适用于平行四边形区域的线性规划模型; Ferreira等^[10]设计了

作者简介: 冯嘉宇(1996-), 男, 硕士研究生, 主要研究方向: 建筑设计自动化、强化学习; 王潇霆(1996-), 男, 硕士研究生, 主要研究方向: 建筑技术自动化、强化学习; 沈 炜(1973-), 男, 博士, 教授, 硕士生导师, 主要研究方向: 云计算、机器学习。

通讯作者: 沈 炜 Email: latitude@126.com.

收稿日期: 2022-04-14

一种车位平行叠放的排布方式,用于优化矩形高密度停车场的车位排布;Timpner 等^[11]提出了停车空间优化模型;Banzhaf 等^[12]通过建立混合整数规划模型,将普通停车场改造为高密度停车场;Nourinejad 等^[13]使用排队论构建了混合整数模型,并通过 benders 分解算法进行车位排布。但是,上述这些研究都集中于地表停车场,而没有考虑障碍物的约束。

对于带障碍物约束的地下车库的排列研究,徐涵喆^[14]等设计了一种基于遗传算法的外圈车位排布启发式算法。黄逸彬等^[15]则通过对不规则停车场区域进行图形分割后,运用粒子群算法进行车位排布。利润等^[16]通过网格化,将原问题区域划分为多个可应用排布算法的子区域,提出了一种基于贪婪分割的地下车位排布算法。但是这些研究者都没有充分利用到每栋楼中承重墙之间的空间,同时车位与边界、障碍物之间存在的夹角也造成了大量的面积浪费。

为了充分利用停车场空间,尽可能最大化车位数量,本文提出了一种根据内部障碍物位置划分为内圈和外圈之后,再对两个区域进行分离处理的启发式自动化排布算法。该算法可以适用于内部障碍物大致平行的任意不带曲边的地下车库;可以通过计算合理排布车位、车位和承重柱的位置;可以将排布结果可视化呈现,辅助人工设计。

1 问题描述

车位排布问题通过输入一个地下车库的外边界和内部所有的障碍物,需要在满足约束条件的情况下,实现车位数量的最大化。

1.1 约束条件

地下车库车位排布的约束条件,在传统排样问题基础上,根据车位排布场景的特殊性,其主要约束条件为:

(1)外边界约束:任意车位模块、车道不得超过外边界所包含的范围。

(2)内部障碍物约束:任意车位模块、车道不得与内部障碍物重叠。

(3)车位与车道分离的约束:车位模块不能与车道重叠。

(4)连通性约束:必须保证每个车位可以驶入车道,所有车道之间必须相互连通。

由于约束条件4中的入口和出口需要在车位排布完成之后由人工确定,故本文只需保证各个车位

模块之间有车道连通即可。

1.2 约束条件的数学模型

地下车库的外边界可以用一个点的集合 $P_{border} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, 构成一个封闭的多边形 S_{border} 。在地下车库内,存在多个多边形表示的内部障碍物 $T_{bar} = \{T_1, T_2, \dots, T_m\}$, 每个内部障碍物也由一个点的集合构成 $S_{bar_1} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ 。

外边界和内部障碍物作为输入条件,满足:

$$T_{bar} \subseteq S_{border} \quad (1)$$

定义 P_m 为车位的集合, R 为车道的集合,则存在如下约束:

$$P_{m_i} \subseteq S_{border}, R_j \subseteq S_{border}, \forall P_{m_i} \in P_m, R_j \in R \quad (2)$$

$$P_{m_i} \cap T_{bar_j} = \emptyset, \forall P_{m_i} \in P_m, T_{bar_j} \in T_{bar} \quad (3)$$

$$R_i \cap T_{bar_j} = \emptyset, \forall R_i \in R, T_{bar_j} \in T_{bar} \quad (4)$$

$$P_{m_i} \cap R_j = \emptyset, \forall P_{m_i} \in P_m, R_j \in R \quad (5)$$

其中,式(2)表示外边界约束;式(3)、式(4)表示内部障碍物约束;式(5)表示车位与车道分离的约束。

车道集合 $R = \{R_1, R_2, \dots, R_m\}$ 是由多个多边形组成的集合。车道的连通需要通过不同车道之间的相交来实现,所以对于所有的车道,需满足:

$$R' \cap C_R R' \neq \emptyset \quad \forall R' \subseteq R, \quad C_R R', R' \neq \emptyset \quad (6)$$

同时每个车位必须有车道相邻接。对于表示任意车位的矩形 $ABCD$, 将其四边向外延长车位宽度 β , 如图1所示。延长后满足:

$$A_1, B_1 \in R_i \text{ or } B_2, C_2 \in R_i \text{ or } C_1, D_1 \in R_i \text{ or } A_2, D_2 \in R_i, \exists R_i \in R \quad (7)$$

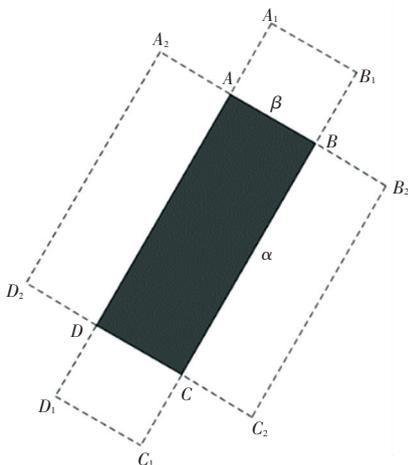


图1 车位向外延长

Fig. 1 Parking Spaces extend outwards

1.3 车位模块

车位模块是由在同一方向上邻接同一车道的车

位排列形成的平行四边形,按照倾角的不同,可以分为 0° 、 30° 、 45° 、 60° 、 90° 和垂直式^[17]。IL Chodash^[18]等人通过分析 0° 、 30° 、 45° 、 60° 、 90° 5种不同角度的停车位在现实停车场中的表现,发现 90° 的车位模块在 67% 的场景中都可以保证车位数量的最大化。赵志强^[19]等用车道面积与车位面积衡量车位实际占用面积,发现 90° 的车位模块占用面积最少。综合这些结论,本文在车位排布中主要采用 90° 的车位模块,在宽度不够时采用 0° 的车位模块。

车位模块中采用每若干个车位之间设置承重柱的方式构建柱网。其中,两个相邻承重柱之间的车位数量取决于承重柱的最大间隔。车位模块中的每个车位都是长为 α 、宽为 β 的矩形,对于承重柱的最大间距 d ,在 90° 的车位模块中需要在每 $\lfloor \frac{d}{\beta} \rfloor$ 个车位之间插入一个承重柱, 0° 的车位模块则为 $\lfloor \frac{d}{\alpha} \rfloor$ 。

1.4 算法流程

本文提出的算法,主要分为以下 4 个阶段:

(1)划分内外区域:根据外边界和内部障碍物的坐标,用矩形包裹所有障碍物后,划分为矩形内部和矩形外部。

(2)矩形内部区域车位排布:通过像素分割栅格化内部区域^[20]并建立矩阵;通过矩阵计算,得到矩形内部区域车位数量最多的排布方案,并保证与

$$A_0(i, j) = \begin{cases} -1, (x_0 + i \cdot \delta, y_0 - j \cdot \delta) \in T_k \\ 0, (x_0 + i \cdot \delta, y_0 - j \cdot \delta) \notin T_k \end{cases}, \forall T_k \in T_{bar}, i \in \left[0, \frac{b}{\delta}\right], j \in \left[0, \frac{a}{\delta}\right], i, j \in N^* \quad (10)$$

根据 A_0 中的元素值,得到 $A(i, j)$, $i \in \left[1, \frac{b}{\delta}\right]$, $j \in \left[1, \frac{c}{\delta}\right]$, $i, j \in N^*$ 。其中:

$$A(i, j) = \begin{cases} -1, A_0(i, j) + A_0(i + 1, j) + A_0(i, j + 1) + A_0(i + 1, j + 1) < 0 \\ 0, A_0(i, j) + A_0(i + 1, j) + A_0(i, j + 1) + A_0(i + 1, j + 1) = 0 \end{cases} \quad (11)$$

矩阵 A 即为描述矩形内部区域的模型。其中 -1 表示有障碍物, 0 表示空旷。

2.2 分割精度选择

在判断每个正方形是否空旷时,因为不包含任何障碍物才认为是空旷的,所以存在一定的面积损失^[19]。例如:对于长 6 m、宽 2.5 m 的停车位来说, δ 为 0.5 m 时每个正方形损失会占单车位 $1/60$ 的面积, δ 为 0.25 m 时占 $1/240$ 。所以 δ 越小,损失的正方形面积也越小。但是,当 δ 缩小时,所需计算次数和内存是以 $O(n^2)$ 复杂度增长,导致计算效率的下降。

外部区域连通。

(3)矩形外部区域车位排布:对矩形外部区域进行分割,使用遗传算法计算得到矩形外部区域车位数量最多的排布方案。

(4)车位坐标计算和可视化:根据车位模块的位置,计算车位坐标。

本文主要对第 1 和第 2 阶段进行研究,第 3 阶段矩形外部区域的车位排布,主要参考徐涵喆^[16]等提出的遗传算法排布方案,第 4 阶段车位模块的坐标则通过遍历计算得到。

2 模型建立

2.1 内部矩阵构建

设定一个长和宽均平行或垂直于内部障碍物的最小矩形 S_{rec} , 使得 $T_{bar} \subseteq S_{rec}$, 则矩形外部区域 $S_{outside}$ 为:

$$S_{outside} = C_N(S_{rec} \cap S_{border}) \quad (8)$$

记 S_{rec} 平行于 x 轴方向的边长为 a , 平行于 y 轴方向的边长为 b , 左上角顶点的坐标为 (x_0, y_0) 。按照分割精度 δ 分割 S_{rec} 上的点,可以得到所有点坐标的集合 P_{rec} :

$$P_{rec} = \{ (x_0 + k_1 \cdot \delta, y_0 - k_2 \cdot \delta) \mid k_1 \in \left[0, \frac{a}{\delta}\right], k_2 \in \left[0, \frac{b}{\delta}\right], k_1, k_2 \in N^* \} \quad (9)$$

构建初始矩阵 A_0 , 其中所有元素满足:

为了选择合适的 δ 值,在考虑整除特性的基础上,选择 1 cm、2 cm、5 cm 3 种不同的分割精度,对 3 张不同的地下车库图纸在 i7 9750H 处理器、16 GB 内存环境下进行了分割实验。根据实验结果,为兼容一般计算机的性能,最终决定 δ 取 5 cm。

2.3 外圈的凹角处理

矩形外部区域 $S_{outside}$ 中不包含任何障碍物,所以只用遗传算法就可以计算。遗传算法的关键在于对 $S_{outside}$ 中外边界点集 P_{border} 的凹角分割策略。参考刘彦鹏等^[21]提出的简单多边形剖分算法,若凹角范

围在 $\left(\pi, \frac{3\pi}{2}\right]$ 内, 采用作垂线的方式切割; 在 $\left(\frac{3\pi}{2}, 2\pi\right)$ 内, 采用作延长线的方式切割。因为每个凹点均存在两条对应边, 所以存在两种不同的切割策略, 可得到 $S_{outside} = \{S_{outside_1}, S_{outside_2}, \dots, S_{outside_m}\}$, $m = 2^n$ 。其中, n 为 P_{border} 凹点的数量。

3 算法设计

3.1 内部矩形中主楼内部空隙排布

定义最小闭包矩形为包裹每栋主楼的且平行于坐标轴的最小矩形。记地下车库中主楼的闭包矩形集合为 $B = \{B_1, B_2, \dots, B_n\}$, 满足:

$$B_i \cap B_j = \emptyset, \forall B_i, B_j \in B, i \neq j \quad (12)$$

$$T_{bar} \subseteq (B_1 \cup B_2 \cup \dots \cup B_n) \quad (13)$$

式(12)表示各个矩形之间不存在重叠, 式(13)表示在这些矩形之外不存在其它障碍物。每个矩形可以包含4个点的点集 $P_B = \{(x_1, y_1), (x_2, y_1), (x_2, y_2), (x_1, y_2) \mid x_1 < x_2, y_1 > y_2\}$ 。每栋楼空隙的车位排列算法流程如下:

输入 矩阵 A , 矩阵左上角元素的坐标 (x_0, y_0) , 主楼的包裹矩形集合 B , 车位长宽 α, β , 主楼数量 n

输出 车位集合 L_1 , 修改后的矩阵 A

```

1  for  $k = 1$  to  $n$  do
2       $t \leftarrow \frac{y_0 - y_1}{\delta}, b \leftarrow \frac{y_0 - y_0}{\delta}, l \leftarrow \frac{x_0 - x_1}{\delta}, r \leftarrow \frac{x_0 - x_2}{\delta}$ 
3       $j \leftarrow 0$ 
4      for  $j = 0$  to  $b - t$  do
5          if ( $A[0: \frac{\alpha}{\sigma}, j: \frac{\beta}{\sigma}]$  为全0矩阵)
6               $L_1 \leftarrow L_1$  中新增车位
7               $A[0: \frac{\alpha}{\sigma}, j: \frac{\beta}{\sigma}] \leftarrow$  全2矩阵
8               $j \leftarrow$  前一车位终止位置+1
9          end if
10     end for
11     for  $j = 0$  to  $l - r$  do
12         if ( $A[0: \frac{\beta}{\sigma}, j: \frac{\alpha}{\sigma}]$  为全0矩阵)
13              $L_1 \leftarrow L_1$  中新增车位
14              $A[0: \frac{\beta}{\sigma}, j: \frac{\alpha}{\sigma}] \leftarrow$  全2矩阵
15              $j \leftarrow$  前一车位终止位置+1

```

```

16         end if
17     end for
18      $A[t; b, l; r] \leftarrow$  全-1矩阵
19 end for
20 return  $L_1, j$ 

```

经该流程处理后, 得到如图2所示的可视化结果。

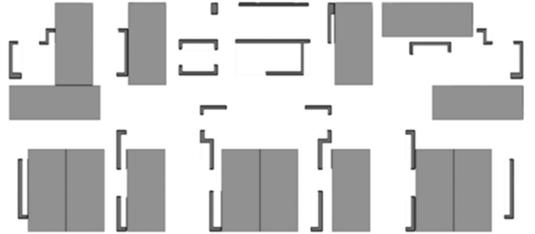


图2 一栋楼内空隙中的车位排布

Fig. 2 The arrangement of parking Spaces in the voids of a building

3.2 内部矩形中主楼内部空隙排布

对于矩阵 A , 给出如下定义:

定义1 若 $Z \subseteq A$ 且 $Z = 0$, 则称 Z 为 A 的全0子矩阵。用全0子矩阵中元素的个数来表示矩阵的大小, 对应矩形的面积。

定义2 若对于 A 的全零子矩阵 Z_m , 不存在另一个全0子矩阵 Z_0 使得 $Z_m \subseteq Z_0$, 则称 Z_m 为 A 的极大全0子矩阵。

定义3 若一个矩形内不包含任何障碍物, 则称其为全空矩形。

定义4 在内部矩形区域内, 若对于一个全空矩形, 不能被任何一个其它的全空矩形完全包裹, 则称其为极大全空矩形。

对于全空矩形内的车位排布, 可以充分利用矩形的直角特性^[22], 采用两排90°的车位模块间隔一条过道的模式依次循环, 在不足时采用0°的车位模块代替, 如图3所示。

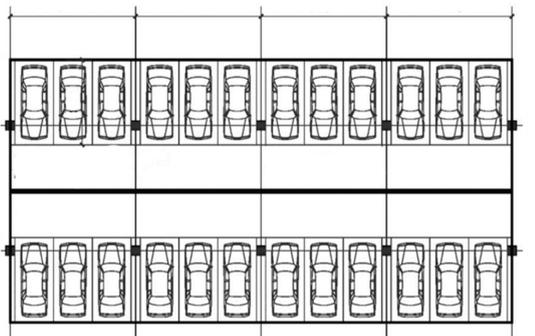


图3 矩形区域内的车位模块设置

Fig. 3 Parking Spaces extend outwards

内部矩形中剩余部分的车位排布算法流程如下:

- 输入** 上一步修改后的矩阵 A , 矩阵左上角元素的坐标 (x_0, y_0) , 车位长宽 α, β , 车位集合 L_1
- 输出** 车位集合 L_2 , 车道集合 R_1
- 1 $Z_{\max} \leftarrow$ 矩阵 A 中最大的极大全 0 子矩阵
 - 2 $r \leftarrow Z_{\max}$ 的行数, $c \leftarrow Z_{\max}$ 的列数
 - 3 L_2, R_1, \leftarrow 空集合
 - 4 While $r \cdot \delta > \alpha + \beta$ && $c \cdot \delta > \alpha + \beta$ do
 - 5 $p \leftarrow$ 横向车位模块排布方案
 - 6 $q \leftarrow$ 纵向车位模块排布方案
 - 7 $p_1, p_2, p_3 \leftarrow p$ 的水平、垂直、中心对称方案
 - 8 $q_1, q_2, q_3 \leftarrow q$ 的水平、垂直、中心对称方案
 - 9 $m_L \leftarrow p, q$ 及其对称方案与 L_1 堵塞判断最后有效车位最大者
 - 10 $m_R \leftarrow m_L$ 对应车道
 - 11 $L_2 \leftarrow m_L \cup L_2$
 - 12 $R_1 \leftarrow m_R \cup R_1$
 - 13 $Z_{\max} \leftarrow$ 全-1 矩阵
 - 14 end While
 - 15 $R_1 \leftarrow R_1$ 中所有道路延长 β
 - 16 return L_2, R_1

3.3 矩形外部排布

矩形外部使用遗传算法对于每一切割区域进行计算,并考虑计算结果造成的 L_1, L_2 中车位堵塞损失,计算得到每种切割方案可排布车位数后,取其中车位数量最多的切割方案。

对于与车道集合 R_1 有重合部分的车位模块,根据车道切割成若干子车位模块之后,计算车位的坐标。最终可以得到矩形外部排布的车位集合 L_3 和车道集合 R_2 。

4 实例验证

为验证基于分离障碍物的地下车库车位排布启发式算法的效果和计算速度,对 5 张不同的地下车库图纸进行计算。采用 4 核、3.40 GHz 处理器、内存为 16 GB 的计算机进行实验,矩形内部区域分割精度 δ 为 5 cm,车道和车位模块长度均为 6 m,承重柱边长 0.5 m,编程语言为 python3.8.1。得到的计算结果见表 1。

上述结果中,算法计算得到的结果总体上优于人工排列的结果。如果考虑后续人工对车道设置的继续优化,仍有较优的结果。其中,图纸 4 算法和人工排布结果如图 4、图 5 所示。

表 1 图纸运算结果

Tab. 1 Drawing Operation Result

编号	车库面积	算法车位数	人工车位数	计算时间
1	24 869	735	710	1 小时 03 分
2	28 991	750	744	1 小时 15 分
3	23 647	662	651	55 分
4	23 175	676	649	47 分
5	19 380	599	583	40 分

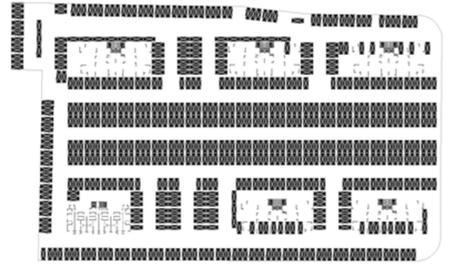


图 4 图纸 4 算法排布结果

Fig. 4 Result of algorithm arrangement in Drawing 4

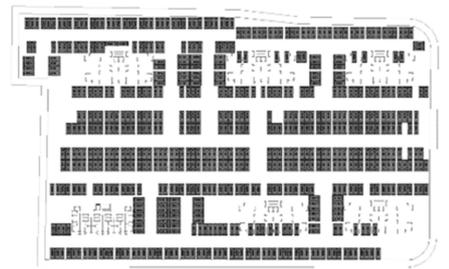


图 5 图纸 4 人工排布结果

Fig. 5 Result of manual arrangement in Drawing 4

5 结束语

本文算法对地下车库按照障碍物进行了区域分割,分别用像素分割和遗传算法两种启发式的求解方法,计算得到了内外区域的排布方案,对比人工排列的实际工程图纸,可以得出如下结论:

(1)通过对矩形内部区域进行像素分割,将复杂的内部障碍物用矩阵元素表示,且充分利用了车位、车位模块、车道中平行和垂直的特性;创造性地利用了每栋楼中的空隙区域,并通过对称、反转每个极大全空矩形中的最优排布方案,保证矩形内部车位总数量的最大化。

(2)通过解决矩形内外结合部区域的连通性问题,可以保证全局的车位数量最大化;比较方便更改车位、道路等长宽参数,可以满足人工对于车位尺寸等的不同需求;该算法可以在较快的时间内完成车位的排布和坐标计算,为人工排列提高效率,增加地

下车库有限区域内的车位数量。

(3)车位的坐标以可视化呈现,为人工排列车位提供一定的参考依据,节省了人工排列的时间。

参考文献

- [1] 李光松. 我国城市停车发展现状问题原因及对策研究——以合肥市肥东县为例 [C]//中国重型机械工业协会停车设备工作委员会:中国重型机械工业协会停车设备工作委员会,2021:133-135.
- [2] 张君博,何坤,董蕴豪,等. 城市地下停车空间的规划布局模式与管理政策优化研究[J]. 现代隧道技术,2020,57(S1):698-704.
- [3] 谷锦彪. 基于驾驶人行为特性的停车位规划方法研究[D]. 合肥:合肥工业大学,2017.
- [4] FOWLER R J, PATERSON M S, TANIMOTO S L. Optimal packing and covering in the plane are NP - complete [J]. Information Processing Letters, 1981, 12(3): 133-137.
- [5] ABDELATAHA A S, TAHA M A. Parking capacity optimization using linear programming [J]. Journal of Traffic and Logistics Engineering, 2014,2(3): 176-181.
- [6] KONG Y, LE VINE S, LIU X B. Capacity impacts and optimal geometry of automated cars' surface parking facilities[J]. Journal of Advanced Transportation, 2018: 1-13.
- [7] HUANG Junhao, LIAO Tianchi, KANG Haining, et al. A general algorithm for rectangular parking optimization design[J]. Journal of Physics Conference Series, 2020, 1486: 032029.
- [8] SYAHRINI I, SUNDARI T, ISKANDAR T, et al. Mathematical model of parking space unit for triangular parking area [C]// 4th International Conference on Operational Research. Lishon: IOP Publishing Ltd, 2018: 012012.
- [9] IHDA Hasbiyati, WIDIWATI Putri, ARISMAN Adnan, et al. Parking lot optimization in parallelogram using the concept area of rectangular and right triangle [J]. Pure and Applied Mathematics Journal, 2019, 8(4): 77-82.
- [10] FERREIRA M, DAMAS L, CONCEIAO H, et al. Self-automated parking lots for autonomous vehicles based on vehicular Ad hoc networking [C]//IEEE Intelligent Vehicles Symposium. Dearborn: IEEE, 2014: 472-479.
- [11] TIMPNER J, FRIEDRICH S, VAN BALEN J. k - stacks: high density valet parking for automated vehicles [C]//Intelligent Vehicles Symposium. Seoul: IEEE, 2015: 895-900.
- [12] BANZHAF H, QUEDENFELD F, NIENHUSER D. High density valet parking using k - deques in driveways [C]//Intelligent Vehicles Symposium. Redondo Beach: IEEE, 2017: 1413-1420.
- [13] NOURINEJAD M, BAHRAMI S, ROORDA M J. Designing parking facilities for autonomous vehicles [J]. Transportation Research Part B Methodological, 2018, 109: 110-127.
- [14] 徐涵喆, 黄逸彬, 杨赫, 等. 基于规则的城市地下车库外圈车位排布启发式算法[J]. 北京邮电大学学报, 2019, 42(4): 102-108.
- [15] 黄逸彬, 杨赫, 周钟秉, 等. 基于图形分割的城市地下车库车位排布优化方法[J]. 北京邮电大学学报, 2020, 43(4): 7-14.
- [16] 利润. 地下停车场车位自动化排布方法研究[D]. 广州: 华南理工大学, 2020.
- [17] 李焕宇, ChungSChen. 大型停车场中汽车的最佳车位角度设计 [J]. 山东交通科技, 1998(2): 79-83.
- [18] CHODASH I L. Relative efficiencies of various parking angles [J]. ITE journal, 1986, 56(3): 34-37.
- [19] 赵志强. 地下机动车库的优化设计与实践[D]. 南京: 中国人民解放军理工大学, 2015.
- [20] 任少杰, 杨正才, 张英泽. 基于栅格化四边形区域与改进人工势场的车辆行驶轨迹研究[J]. 湖北汽车工业学院学报, 2020, 34(3): 6-10.
- [21] 刘彦鹏, 吴明光, 张玉润. 基于 AutoCAD 的简单多边形剖分算法[J]. 计算机工程与应用, 2006(5): 43-45, 49.
- [22] 季君, 邢斐斐, 黄敦华, 等. 一种分阶段式匀质块最优矩形排布方法[J]. 锻压技术, 2021, 46(2): 46-51.
- [3] ZADEH L A. A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination [J]. Ai Magazine, 1986, 7(2): 85-90.
- [4] LIU Z, DEZERT J, QUAN P, et al. Combination of sources of evidence with different discounting factors based on a new dissimilarity measure [J]. Decision Support Systems, 2012, 52(1): 133-141.
- [5] XIAO F. CED: A Distance for Complex Mass Functions [J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, PP(99): 1-11.
- [6] MI X, KANG B. A Modified Approach to Conflict Management From the Perspective of Non-Conflicting Element Set [J]. IEEE Access, 2020, 8: 73111-73126.
- [7] XIAO Fuyuan. An Improved Method for Combining Conflicting Evidences Based on the Similarity Measure and Belief Function Entropy [J]. International Journal of Fuzzy Systems, 2018, 20(4): 1256-1266.
- [8] AN J, MENG H, LI F, et al. A Novel Fuzzy Approach for Combining Uncertain Conflict Evidences in the Dempster-Shafer Theory [J]. IEEE Access, 2019, 7: 7481-7501.
- [9] TAO R, XIAO F. Combine Conflicting Evidence Based on the Belief Entropy and IOWA Operator [J]. IEEE Access, 2019, 7: 120724-120733.
- [10] YAN H, DENG Y. An Improved Belief Entropy in Evidence Theory [J]. IEEE Access, 2020, 8: 57505-57516.
- [11] HUI Z, YONG D. Entropy Measure for Orderable Sets [J]. Information Sciences, 2021, 561: 141-151.
- [12] DENG Y. Deng entropy [J]. Chaos, Solitons and Fractals, 2016, 91: 549-553.
- [13] MURPHY C K. Combining belief functions when evidence conflicts [J]. Decision Support Systems, 2000, 29(1): 1-9.
- [14] YONG D, WENKANG S, ZHENFU Z, et al. Combining belief functions based on distance of evidence [J]. Decision Support Systems, 2004, 38(3): 489-493.

(上接第24页)